Lab tasks – CE0825a

James A Sutherland

Weeks 1-11, Spring 2016

Introduction

The course is structured around 12 weeks. Week 7 is reserved for 'feedback week' and week 12 is to revise and summarise the overall content, with 10 to introduce and practice new material.

The portfolio work is based on the approach used in the web design courses: each week, add some content to your Word document, usually some code you have written and a screenshot or two of it in action. Unlike the full coursework assignment (where you spend multiple weeks developing a single elaborate project to a high standard), the idea is just to show that you can achieve a particular task – for example, reading numbers from the keyboard then calculating their mean.

Each week's lab task will carry 2 points (so the full semester's work will give a score out of 20). As long as your code does the job, you get both points for that week.

1 Week 1 – types, autoboxing, conversions

Explore how Java converts variables from one type to another, both between primitive types (such as double and int) and the corresponding objects (Double and Integer), a conversion known as *autoboxing* and between numerical types.

Note in particular 5.1.11 from the link below: *everything* in Java can be converted into a String. Every Object has a `toString` method for doing that.

In general terms, arithmetic doesn't actually apply to strings: it makes no sense to multiply or divide them. Java makes a special case of addition, using + on Strings as shorthand for appending them; behind the scenes, the compiler actually replaces that + with calls to an `append` method.[1]

Lots of technical detail about type conversion here:

https://docs.oracle.com/javase/specs/jls/se7/html/jls-5.html

Key point 1: If you perform an operation (add, divide etc) on two different types of number (such as an int and a float), which type does the result have and why? (Hint: which option would involve losing information?)

[1]Operating directly on a String is slow, so for performance reasons modern Java will use a faster mechanism such as a StringBuilder. Something you might find useful to know later if you're manipulating a lot of text.

Key point 2: You can't really 'add' anything to a string, so Java will *append* to it instead.[2] So, what happens if you try to add 2 to "2"?

(Note: the original lab assignment asked you to document this by building a table of outcomes for each possibility. For simplicity, just stating the logic behind those results will do, rather than listing everything.)

2 Week 2 – Console I/O, lists, exceptions

Lecture 2 introduced reading from the console (via the `BufferedReader` and `InputStreamReader` classes with `System.in`) and using lists and exceptions.

Key point 1: Read from the console and store the numbers (integer or floating point) in a list, catching any `NumberFormatException` rather than letting it crash your code.

Key point 2: Iterate through your list to calculate the mean (the sum, divided by the number of items). Mathematically speaking, keeping a running total and a counter would give the same result – but that would be maths, not Java list handling!

3 Week 3 – Graphical User Interfaces: SWT

The lecture introduced threading and the origins of the SWT graphical library used by Eclipse. The lab exercise is intended to get the beginnings of a basic SWT programme up and running to build on later: in particular, getting the right SWT library downloaded and connected to your project in Eclipse, then successfully accessing the first few components of SWT.

Learning more about SWT is a good use of your time: we will be using it a lot more later!

http://www.eclipse.org/swt/docs.php

Key point 1: Get an SWT window open.

Key point 2: Add some menu items to it.

4 Week 4 – Maps, Enums and modularisation

The lecture introduces a 'case study', some example code and extensive documentation intended to help you get the hang of Java programming. The example code starts with a very simple map, a grid of `boolean` values (similar to the maps used with Robby the Robot in year 1's Java course).

Key point 1: Get the example code running.

Key point 2: Change the code to use an Enum type with at least three possible values (for example, GRASS, RIVER, TREE) and to print a simple text map on the console.

[2] Useful general knowledge: other languages behave differently: Perl, for example, will convert a string "2" into a number then do arithmetic on that, so "2"+2 gives you the 4 you might expect.

5 Week 5 – Web, Menus

Note: work up as far as the previous week should be uploaded to Blackboard during this week. If necessary, use this lab session to catch up first. Weeks 5-9 will be submitted at the end of week 10.

The lecture demonstrates how to send or retrieve data over the Web, create menu items and handle clicks on them.

For the first task, a slight variation on the lecture content will be needed. You may need to read about the Java `File` class and work out how to copy the bytes out to a file.

For the second task, there are more possibilities than the two shown in lecture examples. Explore how the other options work: submenus, dividing lines within a menu.

Key point 1: Retrieve an *image* from the web and save it to disk when run.

Key point 2: Get menus working in your SWT application from last week, with a web page of your own content loaded locally rather than from the web.

6 Week 6 – Images, Maps

This week covers graphics file formats, and presents the official University campus map – which, unfortunately, is in completely the wrong format for a map: JPEG, designed for photographs not diagrams.

Key point 1: Fetch and display a level of the map in an SWT window.

Key point 2: Provide a way to switch which level is displayed: a menu, a tab per level or another mechanism.

7 Week 7 – Feedback Week

This week will be used to discuss the coursework. Prepare by brainstorming how you will implement a map of the University campus as a Java SWT application: what features will you implement, and how? Prepare a draft specification, and discuss this during the usual lab session timeslot.

This week will also be used to return grades for weeks 1-4 and address any problems encountered in that work.

8 Week 8 – Memory, JNA, Animation

This week gives some background to memory management and introduces Java Native Access, the interface for Java code to call native functions, with an example opening a native MessageBox.

There is also a simple example of animation in SWT using a timed callback to move a circle around the screen.

Key point 1: Call another native Windows function using your own JNA wrapper.

Key point 2: Animate something else in SWT.

9 Week 9 – Regular Expressions, Java SQL

This week explored regular expressions and their use in Java, as well as the Java database interface JDBC.

Key point 1: Read user input and test whether it's a valid staff/student ID or not.

Key point 2: Connect to lochnagar, retrieve and display some data using SQL.

10 Week 10 – Project, Testing, Random

This week gives the final briefing about the project work and marking arrangements, introduces the concept of unit testing and JUnit in particular, then the random number generators.

Key point 1: Write your own test code for some earlier lab work

Key point 2: Add a test that gets a random number and tests it's 42

11 Week 11 – Data Structures, Algorithms, Searching, Soundex

This week introduces some data structures, the $O()$ notation for scalability and the Soundex algorithm for encoding differently spelled homophones alike.

Data structures: hash tables, red-black trees, arrays and linked lists.

Scalability: Constant space, linear time etc.

Soundex: One letter and three digits to encode the *sound* of each word.

Key point 1: Identify Java's built in data structures and how they scale

Key point 2: Implement Soundex correctly, demonstrating that homophones collide

Final Portfolio Submission

Weeks 5-11 inclusive should be submitted through Blackboard as a single PDF or Word document by the end of Monday the 11th of April, the day of the final lecture in week 12. You can also use that lab session for help with this portfolio and the project.

The project is due the following week, with details in a separate document.

Checklist

1. (a) What type is 1 (an int) + 1.2 (a float)? Why?
   (b) What does trying to add the number 2 to a string '2' give you?
2. (a) Read Strings from the user and convert them to numbers
   (b) Store those numbers and print out the mean of them
3. (a) Get an SWT window showing
   (b) Put menu items in it (no need for them to work yet!)
4. (a) Get the example code to compile and run
   (b) Create an Enum with 3+ options, use that instead of booleans
5. (a) Download an image file from the web to disk
   (b) Get your menu items working properly with a local HTML help page
6. (a) Display a level of the Abertay campus map
   (b) Provide a way to switch between levels of that map
7. *Feedback week: plan and discuss coursework, receive weeks 1-4 marks*
8. (a) Call something other than MessageBoxA via JNA
   (b) Animate something other than a circle in SWT
9. (a) Read and validate staff/student IDs
   (b) Retrieve and display data using SQL
10. (a) Write your own test code for some earlier lab work
    (b) Add a test that gets a random number and tests it's 42
11. (a) Find built in Java hash tables etc, write O(...) time for each
    (b) Write and test own Soundex implementation